Säännöllisen kielen tunnistavat Turingin koneet

- Myös säännöllisen kielen hyväksyvien Turingin koneiden tunnistaminen voidaan osoittaa ratkeamattomaksi palauttamalla universaalikielen tunnistaminen tähän ongelmaan

Päätösongelma on:

"Hyväksyykö annettu Turingin kone säännöllisen kielen?"

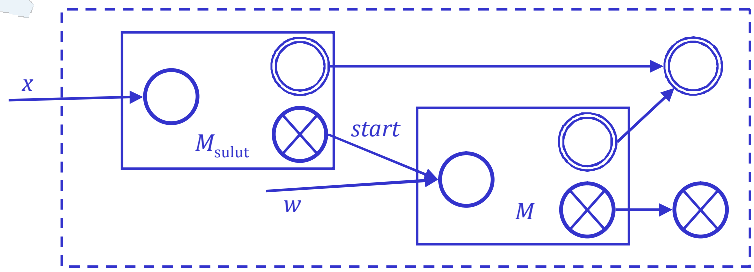
$$\mathrm{REG} = \{ \langle M \rangle \mid M \text{ on Turingin kone ja } L(M) \text{ on säännöllinen} \}$$

Lause (5.3) REG *ei ole rekursiivinen*

Todistus.

- Oletetaan, että REG:lla olisi totaalinen tunnistajakone M^{T}_{REG}

- Tämän avulla voitaisiin muodostaa totaalinen tunnistajakone kielelle U
- Olkoon M mv. Turingin kone, jonka toimintaa syötteellä w tarkastellaan
- Tasapainoisten sulkuparien kieltä vastaava $\{ 0^n1^n \mid n \geq 0 \}$ tunnetusti ei ole säännöllinen, mutta helposti tunnistettavissa Turingin koneella
- Olkoon M_{sulut} ko. kielen totaalinen tunnistaja
- Olkoon M_{ENC} nyt kone joka syötteestä $\langle M, w \rangle$ muodostaa koodin Turingin koneelle M^w, joka syötteellä x
  - ensin toimii kuten M_{sulut} syötteellä x.
  - Jos M_{sulut} hylkää x:n, M^w toimii kuten M syötteellä w.
  - Muuten M^w hyväksyy x:n

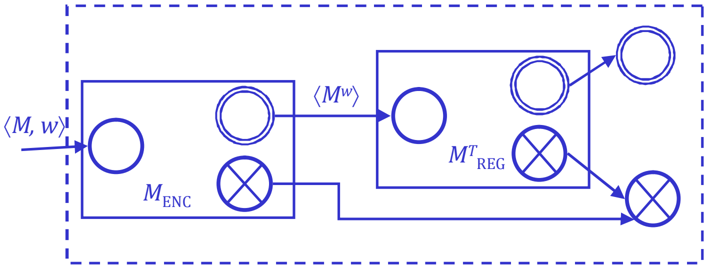


Säännöllisten kielten tunnistaminen: kone M^w

- Kone M^w siis joko hyväksyy säännöllisen kielen $\{0, 1\}^*$ tai ei-säännöllisen $\{0^n1^n \mid n \geq 0\}$
- Merkkijonon w hyväksyminen/hylkääminen koneella M palautuu koneen M^w hyväksymän kielen säännöllisyyteen/ei-säännöllisyyteen

$$L(M^w) = \begin{cases} \{0,1\}^* & \text{jos } w \in L(M) \\ \{0^n1^n \mid n \geq 0\} & \text{jos } w \notin L(M) \end{cases}$$

- Olkoon M_{ENC} kone, joka
  - saa syötteenään Turingin koneen M koodin $\langle M \rangle$ sekä binäärijonon w katenaation, $\langle M, w \rangle$, ja
  - jättää tuloksena nauhalle koneen M^w koodin $\langle M^w \rangle$
- Nyt koneet M_{ENC} ja M^T_{REG} yhdistämällä saataisiin seuraavan kuvan esittämä Turingin kone $M_U{}^T$



Universaalikielen U tunnistava totaalinen Turingin kone M_U^T

- M_U^T on totaalinen jos M^T_{REG} on, ja $L(M_U^T) = U$, koska

$$\begin{aligned} &\langle M, w\rangle \in L(M_U^T) \\ \Leftrightarrow\ &\langle M^w\rangle \in L(M^T_{\text{REG}}) = \text{REG} \\ \Leftrightarrow\ &L(M^w) \text{ on säännöllinen} \\ \Leftrightarrow\ &w \in L(M) \\ \Leftrightarrow\ &\langle M, w\rangle \in U \end{aligned}$$

- Lauseen 5.7 perusteella U ei kuitenkaan ole rekursiivinen, joten koneen M_U^T olemassaolo on ristiriita
- Näin ollen kielellä REG ei voi olla totaalista tunnistajakonetta M^T_{REG}
- Olemme siis todistaneet, että kieli REG ei ole rekursiivinen

□

Ricen lause

- Turingin koneen *semanttiseksi* ominaisuudeksi sanotaan mitä tahansa ominaisuutta, joka riippuu vain koneen tunnistamasta kielestä, ei sen syntaktisesta rakenteesta
- Esim.
  - "M hyväksyy tyhjän syötejonon",
  - "M hyväksyy jonkin syötejonon" (NE),
  - "M hyväksyy äärettömän monta merkkijonoa",
  - "M:n tunnistama kieli on säännöllinen (REG)" jne.
- Jos koneilla M_1 ja M_2 on $L(M_1) = L(M_2)$, niin niillä on täsmälleen samat semanttiset ominaisuudet

- Abstraktimmin: semanttinen ominaisuus S on mikä tahansa kokoelma aakkoston $\{0, 1\}$ RE-kieliä
- Koneella M on ominaisuus S jos $L(M) \in S$.
- *Triviaalit* ominaisuudet ovat $S = \emptyset$ ja $S = \mathrm{RE}$
- Ominaisuus S on *ratkeava*, jos kieli

$$\mathrm{codes}(S) = \{ \langle M \rangle \mid L(M) \in S \}$$

  on rekursiivinen.

Ricen lause *Kaikki Turingin koneiden epätriviaalit semanttiset ominaisuudet ovat ratkeamattomia*

Laskentahistoriat

- Turingin koneen laskentahistoria on niiden tilanteiden jono, joihin kone joutuu syötettä käsitellessään
- Koneen M hyväksyvä laskentahistoria syötteellä w on tilanteiden jono $C_1, C_2, \ldots, C_l$, missä
  - C_1 on alkutilanne $q_0\, w$,
  - C_l on hyväksyvä tilanne ja
  - kukin C_{i-1} johtaa suoraan C_i:hin
- Vastaavasti määritellään hylkäävä laskentahistoria
- Laskentahistoriat ovat äärellisiä jonoja —
  jos M ei pysähdy syötteellä w, niin hyväksyvää tai hylkäävää historiaa ei ole

Lineaarisesti rajoitetut automaatit

- Lineaarisesti rajoitettu automaatti (LRA) on Turingin kone, joka ei voi ottaa lisää työtilaa käyttöönsä
- Kone voi käyttää vain syötemerkkijonon pituuden verran työtilaa
- Koska nauha-aakkosto kuitenkin voi olla syöteaakkostoa laajempi, voi LRA käyttää vakiokertoimella korotetun muistimäärän
- Kontekstisten kielten ratkaisijoita
- Jos LRA:n
  - tilojen määrä on q,
  - nauha-aakkoston merkkien määrä g ja
  - syötteen pituus n,

  niin sen mahdollisten tilanteiden lukumäärä on $q \cdot n \cdot g^n$

Lause (5.9)
Lineaarisesti rajoitettujen automaattien hyväksymisongelma
$A_{\mathrm{LRA}} = \{ \langle M, w\rangle \mid M \text{ on LRA ja hyväksyy merkkijonon } w \}$
on ratkeava

Todistus. Jos LRA laskentansa kuluessa päätyy uudelleen samaan tilanteeseen kuin missä se jo kerran oli, niin laskennan jatkuessa näin tulee käymään yhä uudelleen. Eli tällöin LRA:n laskenta on silmukassa eikä tule pysähtymään.

Koska LRA:lla on $q{\cdot}n{\cdot}g^n$ mahdollista tilannetta, niin jos laskenta ei ole päättynyt yhtä monen askelen kuluessa, on se silmukassa.
A_{LRA}:n ratkaisemiseksi riittää siis simuloida syötteellä w koneen M toimintaa $q{\cdot}n{\cdot}g^n$ askelta tai kunnes laskenta pysähtyy.
□

Lause 5.10 *Lineaarisesti rajoitettujen automaattien tyhjyysongelma*
$E_{\mathrm{LRA}} = \{ \langle M\rangle \mid M \text{ on LRA ja } L(M) = \emptyset \}$
on ratkeamaton

Todistus. Palautus ratkeamattomasta Turingin koneiden hyväksymisongelmasta (universaalikielestä).
Vastaoletus: E_{LRA} on ratkeava, eli on olemassa totaalinen Turingin kone $M_E{}^T$ sen tunnistamiseksi.

Olkoon M mv. Turingin kone, jonka toimintaa syötteellä w tarkastellaan. Muodostetaan LRA B, joka tunnistaa kaikki M:n hyväksyvät laskennat merkkijonolla w.

Tällöin saamme palautettua Turingin koneiden hyväksymisongelman LRA:iden tyhjyysongelmaan:

$$\begin{cases} L(B) \neq \emptyset & \text{jos } w \in L(M) \\ L(B) = \emptyset & \text{jos } w \notin L(M) \end{cases}$$

Koneen B tulee siis hyväksyä syöte x mikäli se on M:n hyväksyvä laskentahistoria syötteellä w. Olkoon syötteen koodaus $x = C_1\#C_2\#\cdots\#C_l$.

B tarkastaa, että x täyttää hyväksyvän laskentahistorian ehdot:

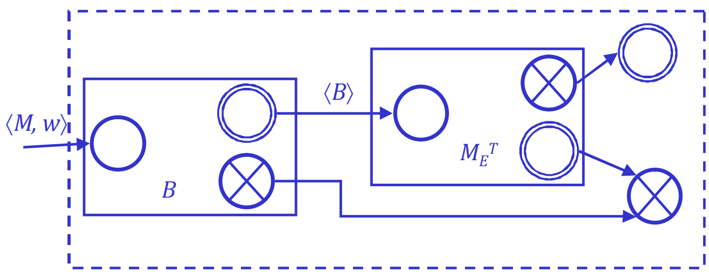
- $C_1 = q_0\, w$,
- C_l on hyväksyvä tilanne, s.o. **`accept`** on tilanteen C_l sisältämä tila ja
- $C_{i\text{-}1} \succ_M C_i$:
  - tilanteiden $C_{i\text{-}1}$ ja C_i tulee olla muilta osin ekvivalentit, paitsi aiemman nauhapään position kohdalta ja viereisissä muistipaikoissa sekä
  - tehtyjen muutosten tulee noudattaa M:n siirtymäfunktiota.

Annettuna M ja w on mahdollista muodostaa B automaattisesti.

Nyt yhdistämällä koneet B ja M_E^T seuraavan kuvan osoittamalla tavalla, saadaan totaalinen tunnistaja Turingin koneiden hyväksymisongelmaan (universaalikielelle).

$$
\begin{aligned}
&\langle M, w\rangle \in L(M_U^T) \\
&\Leftrightarrow \langle B\rangle \notin L(M_E^T) \\
&\Leftrightarrow L(B) \neq \emptyset \\
&\Leftrightarrow w \in L(M) \\
&\Leftrightarrow \langle M, w\rangle \in U
\end{aligned}
$$

Tämä on ristiriita, joten kieli E_{LRA} ei voi olla ratkeava. □



Universaalikielen U tunnistava totaalinen Turingin kone M_U^T

6.1 Rekursiiviset palautukset

- Olkoon $M = (Q, \Sigma, \Gamma, \delta, q_0,$ **accept**, **reject**$)$ mv. standardi-mallinen Turingin kone
- Määritellään koneen laskema *osittaisfunktio* $f_M\colon \Sigma^* \rightarrow \Gamma^*$ seur.

$$f_M(w) = \begin{cases} u, & \text{jos } q_0\, w >>>_M u\, q\, av, \\ & q \in \{\text{accept, reject}\}, av \in \Gamma^* \\ \text{määrittelemätön}, & \text{muuten} \end{cases}$$

- Kone M siis kuvaa merkkijonon $w \in \Sigma^*$ koneen nauhapään vasemmalla puolella sijaitsevalle merkkijonolle u, jos laskenta pysähtyy syötteellä w
- Jos laskenta ei pysähdy, funktion arvoa pisteessä w ei ole määritelty

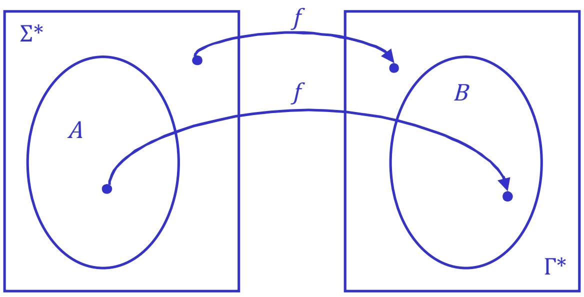
- Osittaisfunktio $f\colon \Sigma^* \rightarrow A$ on osittaisrekursiivinen, jos
  $\exists\, M\colon f = f_M, A \subseteq \Gamma^*$
- Osittaisrekursiivifunktio f on rekursiivinen, jos se voidaan laskea totaalisella Turingin koneella. Ts. jos sen arvo $f(w)$ on määritelty kaikilla w
- Formalisoidaan ajatus, että ongelma A on "enintään yhtä vaikea" kuin ongelma B seuraavasti
  - Olkoot $A \subseteq \Sigma^*$, $B \subseteq \Gamma^*$ formaaleja kieliä
  - A voidaan palauttaa rekursiivisesti B:hen, merkitään

$$A \leq_m B,$$

jos on olemassa rekursiivinen funktio $f\colon \Sigma^* \rightarrow \Gamma^*$ s.e.

$$w \in A \Leftrightarrow f(w) \in B \qquad \forall\, w \in \Sigma^*$$

- Tällöin A:n tapaus w voidaan muuttaa rekursiivisesti B:n tapaukseksi $f(w)$ ja
  "onko w:llä ominaisuus A?" ≡
  "onko $f(w)$:llä ominaisuus B?"



Lemma 5.22 *Kaikilla kielillä A, B, C pätee*

i. $A \leq_m A$, (refleksiivisyys)

ii. jos $A \leq_m B$ ja $B \leq_m C$, niin $A \leq_m C$, (transitiivisuus)

iii. jos $A \leq_m B$ ja B on RE-kieli, niin A on RE-kieli ja

iv. jos $A \leq_m B$ ja B on rekursiivinen, niin A on rekursiivinen

Todistus.

i. Valitaan palautusfunktioksi $f(x) = x$.

ii. Olk. f palautusfunktio A:sta B:hen ja g B:stä C:hen.
Merk. $f: A \leq_m B$, $g: B \leq_m C$.

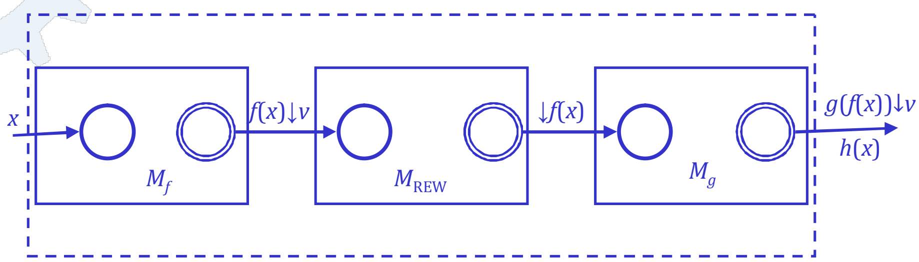
Osoitetaan, että yhdistetty funktio h, $h(x) = g(f(x))$ on palautus $h: A \leq_m C$.

1. h on rekursiivinen: Olkoot M_f ja M_g f:n ja g:n laskevat totaaliset Turingin koneet. M_{REW} korvaa kaikki nauhapään oikealla puolella olevat merkit $\sqcup$:lla ja vie nauhapään nauhan alkuun. Seuraavan kuvan totaalinen kone laskee funktion h.
2. h on palautus:

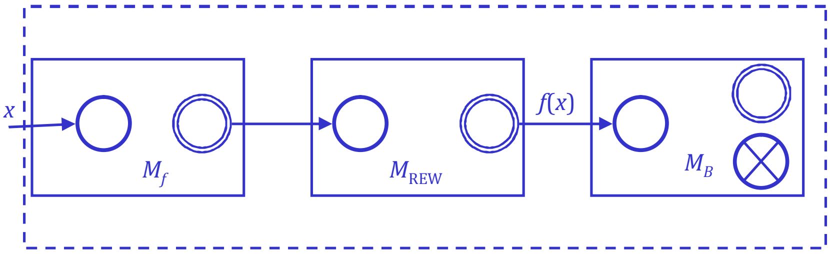
$$\begin{aligned} x \in A \quad &\Leftrightarrow f(x) \in B \\ &\Leftrightarrow g(f(x)) = h(x) \in C, \end{aligned}$$

joten $h: A \leq_m C$.

iii. (ja iv.) Olk. $f: A \leq_m B$, M_B kielen B tunnistajakone ja M_f funktion f laskeva kone. Seuraavan kuvan kone tunnistaa kielen A ja on totaalinen jos M_B on. □



Yhdistetyn kuvauksen laskeva Turingin kone



Kielen A tunnistava Turingin kone

- Aiemmin jo käytimme lauseen 5.22 seurausta ratkeamattomuuden todistamiseen

Seuraus 5.23 *Jos $A \leq_m B$ ja A on ratkeamaton, niin myös B on ratkeamaton*.

Kieli $B \subseteq \{ 0, 1 \}^*$ on RE-täydellinen, jos

1. $B \in$ RE ja
2. $A \leq_m B$ kaikilla $A \in$ RE

Lause *Kieli U on RE-täydellinen*

Todistus. Tiedämme, että $U \in$ RE. Olkoon B mv. RE-kieli. Olkoon edelleen $B = L(M_B)$. Nyt B voidaan palauttaa U:hun funktiolla $f(x) = \langle M_B, x \rangle$. Funktio on selvästi rekursiivinen, ja sille pätee

$$x \in B = L(M_B) \Leftrightarrow f(x) = \langle M_B, x \rangle \in U. \quad \square$$

Lause 6.1 *Olkoon A RE-täydellinen kieli, $B \in$ RE ja $A \leq_m B$. Tällöin myös B on RE-täydellinen.*

- Kaikki "luonnolliset" luokan RE \ REC kielet ovat RE-täydellisiä, mutta kuitenkin siinä on muitakin kieliä
- Luokka RE ei ole suljettu komplementoinnin suhteen, joten sillä on duaaliluokka

$$\text{co-RE} = \{ \bar{A} \mid A \in \text{RE} \}$$

- RE ∩ co-RE = REC (Lause 5.3)
- $B \subseteq \{ 0, 1 \}^*$ on co-RE-täydellinen, jos $B \in$ co-RE ja $A \leq_m B$ kaikilla $A \in$ co-RE
- Kieli A on co-RE-täydellinen, jos ja vain jos kieli $\bar{A}$ on RE-täydell. i
- Kieli $\text{TOT} = \{ \langle M \rangle \mid M \text{ pysähtyy kaikilla syötteillä} \}$

ei kuulu luokkaan RE eikä co-RE